| SEARCH IEEE | SHOP | WEB ACCOUNT

Welcome
United States Patent and Trademark Office

Help FAQ Terms Quick Links » Abstract Plus
IEEE Peer Review

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (116 KB)] PREVIOUS NEXT DOWNLOAD CITATION

Sensitivity analysis for object recognition from large structural libraries

Huet, B. Hancock, E.R.
Dept. of Comput. Sci., York Univ.;

This paper appears in: **Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on**
Meeting Date: 09/20/1999 -09/27/1999
Publication Date: 1999
Location: Kerkyra , Greece
On page(s): 1137-1143 vol.2
Volume: 2, References Cited: 23
Number of Pages: 2 vol.xxvii+1258
INSPEC Accession Number: 6365384

Abstract:
The paper studies the structural sensitivity of line pattern recognition using shape graphs. We compare the recognition performance for four different algorithms. Each algorithm uses a set of pairwise geometric attributes and a neighbourhood graph to represent the structure of the line patterns. The first algorithm uses a pairwise geometric histogram, the second uses a relational histogram on the edges of the shape graph, the third compares the set of attributes on the edges of the shape graph and the final algorithm compares the arrangement of line correspondences using graph matching. The different algorithms are compared under line deletion, line addition, line fragmentation and line end point measurement errors. It is the graph matching algorithm which proves to be the most effective

Index Terms:
computational geometry graph theory image matching object recognition graph matching large structural libraries line addition line correspondences line deletion line end point measurement errors line fragmentation line pattern recognition neighbourhood graph object recognition pairwise geometric attributes pairwise geometric histogram recognition performance relational histogram sensitivity analysis shape graph shape graphs structural sensitivity

Documents that cite this document
Select link to view other documents in the database that cite this one.

Sensitivity Analysis for Object Recognition from Large Structural Libraries

Benoit Huet and Edwin R. Hancock
Department of Computer Science, University of York
York, YO10 5DD, UK
huetb@cs.york.ac.uk erh@cs.york.ac.uk

Abstract

This paper studies the structural sensitivity of line-pattern recognition using shape-graphs. We compare the recognition performance for four different algorithms. Each algorithm uses a set of pairwise geometric attributes and a neighbourhood graph to represent the structure of the line patterns. The first algorithm uses a pairwise geometric histogram, the second uses a relational histogram on the edges of the shape graph, the third compares the set of attributes on the edges of the shape graph and the final algorithm compares the arrangement of line correspondences using graph-matching. The different algorithms are compared under line deletion, line addition, line fragmentation and line end-point measurement errors. It is the graph-matching algorithm which proves to be the most effective.

1 Introduction

Graphs have proved to be seductive yet highly elusive as representations of shape in computer vision [2, 18, 22]. They are seductive since they convey the topological arrangements of features or object primitives in a manner which can be invariant to both viewing angle and shape deformation. On the other hand, they have proved to be very elusive since the process of eliciting a relational abstraction from poor image data is one of extreme fragility [22]. The idea of using graphs as representations of 2D scenes can be traced back to Barrow and Popplestone [2]. Over the past two and half decades, the topic has been the subject of sustained activity. However, the fact remains that inexact graph matching is a difficult problem and that there is no "free lunch" on offer. There is a stark choice of either accepting limited recognition performance or deploying considerable computational resources to get an acceptable answer. This trade-off has been highlighted by the interest in large image databases. If such databases are to be queried using high-level relational object descriptions rather than using low level feature characteristics, then a fast and reliable means of graph-matching is required.

There are several recent examples of the use of graph-based representations for 2D shape recognition. For instance, the FORMS system of Zhu and Yuille [23] uses skeletal shape graphs to model articulated objects. Liu and Geiger [13] have taken these ideas further by developing a hierarchical model which achieves a degree of unification between the articulated shape graph and the detection of raw image features via the Mumford-Shah functional [14]. Amit and Kong [1] have a MAP framework for modelling 2D deformable shape using a decomposable graph representation. More recently, there have been several attempts to use graph retrieval as a means of recognising 2D shapes from databases. Much of this work can be viewed as providing a concrete realisation of the ideas introduced by Leyton's [12] process grammar for shape. For instance Siddiqi *et al* [20] have used the shock graph derived from the singularities of the reaction-diffusion equation to provide a skeletal representation of 2D binary shapes. Shape recognition is realised using the subtree matching algorithm of Reyner [16]. The matching process has been refined by Pelillo *et al* [15] who establish a mean of matching association trees using a relaxation algorithm to find the maximal clique.

The observation underpinning this paper is that although this work has done much to establish the representational expediency of shape graphs, there has been little effort directed at answering the question of how much structural corruption can be tolerated and how recognition accuracy is traded against the computational resources expended. It is these issues which are the focus of this paper. We aim to compare four different strategies for rapid graph-matching. The matching algorithms become increasingly complex by placing greater reliance on graph-structure in the shape recognition process. The application vehicle used in our recognition experiments is a demanding one. We aim to recognise large line-patterns segmented from grey-scale images. The simplest line-matching algorithm uses a histogram of pairwise geometric attributes on pairs of line-segments. The second algorithm uses the edge-set of

a shape-graph to gate contributions to the histogram. The third method uses the similarity of features-sets defined on the graph-edges. The final algorithm uses an iterative process to recover an explicit arrangement of correspondences between graphs and retrieves the pattern from the database on the basis of maximum consistency.

Our study is conducted on a database of some 2500 line patterns. In order to investigate the structural and attribute sensitivity of our matching algorithms we subject the patterns in the database to four different kinds of segmentation error. These are line addition, line deletion, line-splitting and segment end-point measurement errors. We measure the recognition performance against each of these errors. Although the sensitivity pattern is quite subtle, the main conclusion of our study is that the iterative graph-matching algorithm offers the best overall performance.

2 Framework

Formally our recognition problem is posed as follows. We abstract the shapes to be recognised as attributed relational graphs (ARG's). Each ARG in the database is a triple, $G = (V_G, E_G, A_G)$, where V_G is the set of vertices (nodes), E_G is the edge set ($E_G \subset V_G \times V_G$), and A_G is the set of node attributes. In our experimental example, the nodes represent line-structures segmented from 2D images. The edges are established by computing the N-nearest neighbour graph for the line-centres. Each node $j \in V$ is characterised by a vector of attributes, $\underline{x}_j$ and hence $A_G = \{\underline{x}_j | j \in V\}$. In the work reported here the attribute-vector represents the contents of a normalised pairwise attribute histogram.

The database of line-patterns is represented by the set of ARG's $\mathcal{D} = \{G\}$. The goal is to retrieve from the database $\mathcal{D}$, the individual ARG that most closely resembles a query pattern $Q = (V_Q, E_Q, A_Q)$. We now furnish some details of the shape retrieval task used in our experimental evaluation of the recognition method. In particular, we focus on the problem of recognising 2D line patterns in a manner which is invariant to rotation, translation and scale. The raw

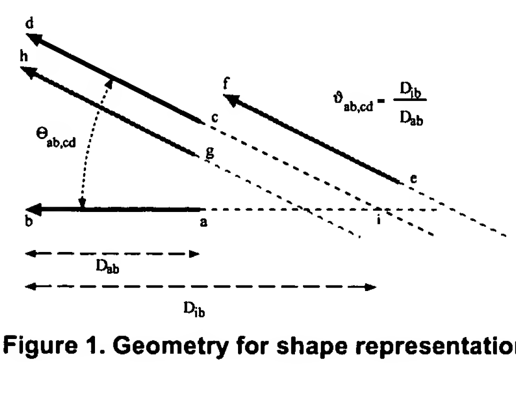


Figure 1. Geometry for shape representation

information available for each line segment are its orientation (angle with respect to the horizontal axis) and its length (see Figure 1). To illustrate how the Euclidean invariant pairwise feature attributes are computed, suppose that we denote the line segments associated with the nodes indexed a and b by the vectors $\underline{v}_a$ and $\underline{v}_b$ respectively. We use two pairwise attributes. The first is the relative angle given by

$$\theta_{a,b} = \arccos\left[\frac{\underline{v}_a \cdot \underline{v}_b}{|\underline{v}_a||\underline{v}_b|}\right]$$

The second is the normalised length ratio between the oriented baseline vector $\underline{v}_a$ and the vector $\underline{v}'$ joining the end (b) of the baseline segment (ab) to the intersection of the segment pair (cd).

$$\vartheta_{a,b} = \frac{1}{\frac{1}{2} + \frac{D_{ib}}{D_{ab}}}$$

The two attributes are used as a feature-vector $z_{a,b} = (\theta_{a,b}, \vartheta_{a,b})^T$ for the line-segment pair.

Each node in the shape graph, i.e. each line in the pattern, is represented by the histogram of its pairwise geometric attributes to the remaining lines in the pattern. This histogram can be thought of as a local estimate of the probability distribution for the pairwise attributes. Accordingly, the angle and position attributes $\theta_{a,b}$ and $\vartheta_{a,b}$ are binned in a histogram. Suppose that $S_a(\mu, \nu) = \{(a,b) | \theta_{a,b} \in A_\mu \wedge \vartheta_{a,b} \in R_\nu \wedge b \in V_D\}$ is the set of nodes whose pairwise geometric attributes with the node a are spanned by the range of directed relative angles A_μ and the relative position attribute range R_ν. The contents of the histogram bin spanning the two attribute ranges is given by $H_a(\mu, \nu) = |S_a(\mu, \nu)|$. Each histogram contains n_A relative angle bins and n_R length ratio bins. The normalised geometric histogram bin-entries are computed as follows

$$h_a(\mu, \nu) = \frac{H_a(\mu, \nu)}{\sum_{\mu'=1}^{n_A} \sum_{\nu'=1}^{n_R} H_a(\mu, \nu)}$$

In the remainder of this paper, we use the notation h^G to denote the normalised histogram for the graph G from the database and h^Q to denote the query histogram.

3 Graph Retrieval Algorithms

The aim in this paper is to compare four graph-based recognition algorithms for retrieving from the database the pattern that most closely resembles a query. The four algorithms use increasingly complex representations. The most straightforward uses a global histogram of pairwise geometric attributes [8]. The next method refines this idea by using only the attributes on the edges of a nearest neighbour graph [10]. The third algorithm uses the set of attributes

on the edges and realises comparison on an element-by-element basis using a robust error kernel [9]. Finally, we use a graph-matching algorithm. Here the similarity of the local attribute histograms is used to provide explicit node correspondences [7]. The matching algorithm iteratively modifies the pattern of correspondences to optimise the *a posteriori* probability of the library patterns given the query pattern. Retrieval is realised by identifying the line-pattern of largest matching probability.

3.1 Attribute Histograms

We commence by considering how pairwise geometric histograms can be used for the purposes of recognition [21]. The idea here is to conglomerate the node histograms into a global histogram [8]. This histogram provides a statistical summary for the pairwise attributes residing on the edges of a nearest neighbour graph. The normalised histogram bin-contents is given by

$$\hat{h}_T^G(\mu, nu) = \frac{\sum_{a \in V_G} H_a(\mu, \nu)}{\sum_{a \in V_G} \sum_{\mu'=1}^{n_A} \sum_{\nu'=1}^{n_R} H_a(\mu, \nu)}$$

The best-matching pattern is retrieved from the database on the basis of similarity with the query pattern histogram. Our similarity measure is the histogram correlation. The measure of pattern correlation is the Bhattacharyya distance. The class identity of the retrieved pattern is

$$\omega_Q = \arg\max_{G \in \mathcal{D}} \ln \sum_{\mu=1}^{n_A} \sum_{\nu=1}^{n_R} \sqrt{h_T^Q(\mu, \nu) \times h_T^G(\mu, \nu)}$$

This is the most efficient of our retrieval algorithms. Once the histograms have been pre-compiled and normalised, then the computational overheads are purely related to the number of bin comparisons that must be performed.

3.2 Relational Histograms

The next step is to consider how graph-structure can be used to compute a compact relational summary [6, 19, 3, 10]. This method represents a refinement of the global attribute histogram. Rather then binning the geometric attributes over all node-pairs, we restrict ourselfs to pairs of nodes that are connected by edges in the shape graph. To do this, we modify the set of attributes used to represent the node a to be those that belong to the first-order graph neighbourhood. In other words, we now have $S_a(\mu, \nu) = \{(a, b) | \theta_{a,b} \in A_\mu \wedge \vartheta_{a,b} \in R_\nu \wedge (a, b) \in E_D\}$. The normalised global histogram can now be computed in the way described in the previous subsection of this paper. The effect is to use the edges of the shape-graph to gate contributions to the histogram.

The run-time computation is identical to that for the attribute histogram. However, the amount of precompilation required can be less than that for the attribute histogram. The reason for this is that once the nearest neighbour graph is computed, then fewer angle and relative length attributes need to be computed.

3.3 Feature-sets

The next step is to consider how set-based representation can be used for the purposes of recognition [9]. Here we use a variant of Rucklidge's [17] idea of effecting object recognition by using a variant of the Hausdorff distance to compare a set of features. We aim to effect retrieval on the basis of the similarity of the set of attributes residing on the edges of the nearest neighbour graph. The Hausdorff distance provides a means of comparing a set of unordered observations without having to establish explicit correspondences between individual elements. However, this measure is notoriously susceptible to measurement outliers. For this reason, we choose instead to gauge similarity using a robust error kernel [9]. The class identity of the retrieved pattern is

$$\omega_Q = \arg\max_{G \in \mathcal{D}} \sum_{(i,j) \in E_G} \max_{(I,J) \in E_Q} \left(\Gamma_\sigma(||\underline{z}_{I,J}^Q - \underline{z}_{i,j}^G||) \right)$$

where $\Gamma_\sigma(\rho) = \exp\left(-\frac{\rho^2}{2\sigma^2}\right)$ is a robust weighting kernel, $z_{I,J}^Q$ is a pairwise feature-vector from the query graph and $z_{i,j}^G$ is a feature-vector from a target graph.

This recognition method is more computationally demanding than either of the histogram-based methods since it requires each of the set of attributes to be compared. When compared with the histogram-based methods, the computational effort required is increased by a factor which is proportional to the average bin contents of the histograms.

3.4 Graph Matching

The feature-based similarity measure does not utilise any information concerning the consistency of the arrangement of correspondences between the individual elements. In order to exploit the consistency of correspondences, we use a simplification of the graph-matching scheme developed by Finch, Wilson and Hancock [5]. This poses the retrieval process as one of associating with the query the graph from the database that has the largest *a posteriori* probability of match. In other words, the class identity of the graph which most closely corresponds to the query is

$$\omega_Q = \arg\max_{G' \in \mathcal{D}} P(G'|Q)$$

However, since we wish to make a detailed structural comparison of the graphs, rather than comparing their overall

statistical properties, we must first establish a set of best-match correspondences between each ARG in the database and the query Q. At iteration n the set of correspondences between the query Q and the ARG G is a relation $f_G^n : V_G \mapsto V_Q$ over the vertex sets of the two graphs. The mapping function consists of a set of Cartesian pairings between the nodes of the two graphs, i.e. $f_G^n = \{(a, \alpha); a \in V_G, \alpha \in V_Q\} \subseteq V_G \times V_Q$. The retrieved pattern is the one which has the most consistent pattern of correspondences and satisfies the condition

$$\omega_Q = \arg \max_{G' \in \mathcal{D}} \max_{f_{G'}} P(f_{G'}|G', Q)$$

The pattern of correspondences is assigned to satisfy the following *maximum a posteriori* probability condition

$$f_G^n(a) = \arg \max_{\alpha \in V_Q} p(\underline{x}_a, \underline{x}_\alpha | f_G^n(a) = \alpha) P(f_G^n | E_G, E_Q)$$

Suppose that we use the notation

$$s_{a,\alpha}^n = \begin{cases} 1 & \text{if } f_G^n(a) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

to represent the correspondence assignment. The consistency of global match against the query pattern can be improved by iterating the assignment condition

$$f_G^n(a) = \arg \max_{\alpha \in V_Q} \Bigg[\ln p(\underline{x}_a, \underline{x}_\alpha | f_G^n(a) = \alpha) + \sum_{(a,b) \in E_G} \sum_{(\alpha,\beta) \in E_Q} \Big\{ \ln(1 - P_e) s_{a,\alpha}^{n-1} s_{b,\beta}^{n-1} + \ln P_e (1 - s_{a,\alpha}^{n-1} s_{b,\beta}^{n-1}) \Big\} \Bigg]$$

The probability of match between the pattern-vectors is computed using the Bhattacharyya coefficient between the normalised histograms.

$$P(f_G^n(a) = \alpha | \underline{x}_a, \underline{x}_\alpha) = \exp[-B_{a,\alpha}] = \frac{\sum_{\mu=1}^{n_A} \sum_{\nu=1}^{n_R} \sqrt{h_a(\mu,\nu) h_\alpha(\mu,\nu)}}{\sum_{\alpha \in V_Q} \sum_{\mu'=1}^{n_A} \sum_{\nu'=1}^{n_R} \sqrt{h_a(\mu,\nu) h_\alpha(\mu,\nu)}}$$

With this modelling ingredient, and using the correspondence matches delivered by the graph-matching scheme, the condition for recognition is

$$\omega_Q = \arg \max_{G' \in \mathcal{D}} \sum_{(a,b) \in E'_G} \sum_{(\alpha,\beta) \in E_Q} \Big\{ -B_{a,\alpha} - B_{b,\beta} + \ln(1 - P_e) s_{a,\alpha}^n s_{b,\beta}^n + \ln P_e (1 - s_{a,\alpha}^n s_{b,\beta}^n) \Big\}$$

This is the most computationally demanding of the methods. We have found it to be 10 times slower than the feature-set method and over a 1000 times slower than the histogram-based method.

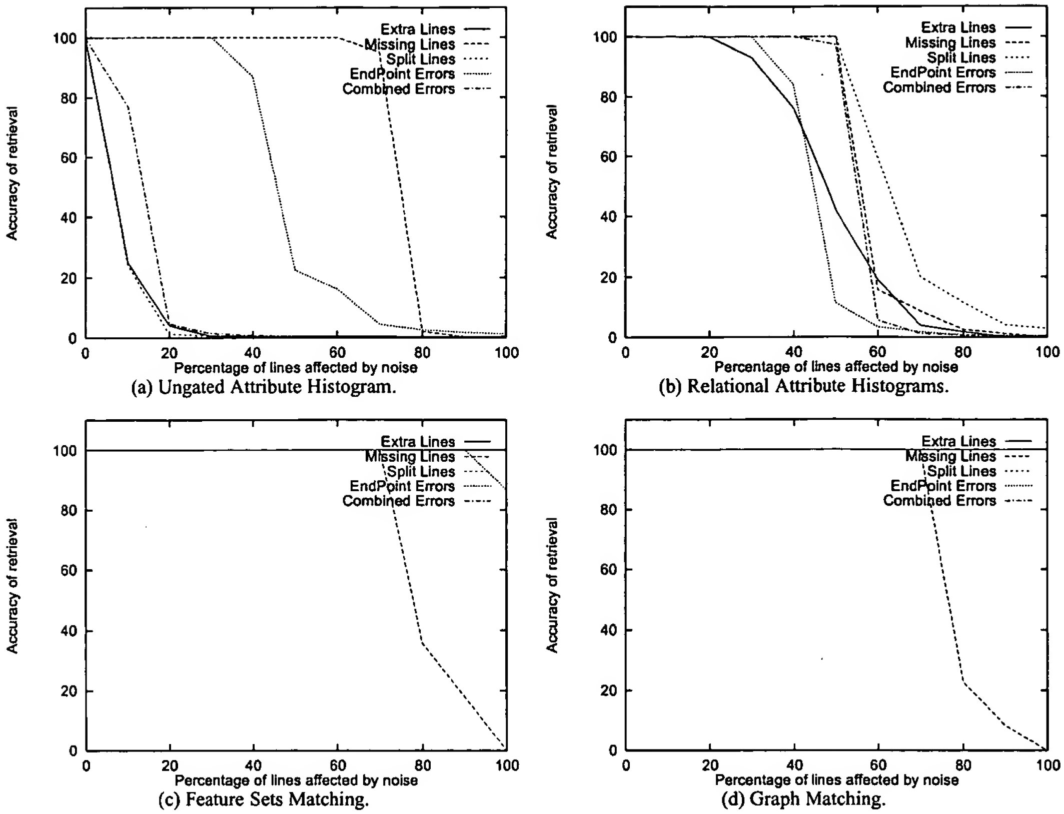
4 Sensitivity Analysis

The aim in this section is to investigate the sensitivity of the four graph-based retrieval strategies to the systematics of the line-segmentation process. To this end we have simulated the segmentation errors that can occur when line-segments are extracted from realistic image data. Specifically, the different processes that we have investigated are listed below:

- **Extra lines:** Here we have added additional lines at random locations. The lengths and angles of the added lines have been generated by randomly sampling the distribution for the existing image-segments.
- **Missing lines:** Here we have deleted a known fraction of line-segments at random locations.
- **Split lines:** Here a predefined fraction of lines have been split into two segments. The splitting process is effected by deleting an internal fraction of each line-segment. The deleted segment is randomly positioned along the line. The fraction of the line deleted is uniformly sampled from the range $(0, 1)$.
- **Segment end-point errors:** Here we have introduced random displacements in the end-point positions for a predefined fraction of lines. The distribution of end-point errors is Gaussian. The degree of error is controlled by the variance of the Gaussian distribution.
- **Combined errors:** Here we have introduced the four different segment errors described above in equal proportion.

The performance measure used in our studies is computed as follows. We query the database with a sample of line patterns. For each pattern in turn we determine whether or not the correct retrieval occurs in the top-ranked position. By computing the fraction of queries that return a correctly recognised recall, we determine the average retrieval accuracy.

We have conducted our experiments with both exact and inexact queries. In the former case an exact match to the line pattern exists prior to the addition of noise. In the latter case, the query pattern is a distorted version of the target in the database. We commence by comparing the overall sensitivity pattern for each of the recognition schemes when exact and inexact queries are performed. Figure 2 shows the sensitivity plots for the case of exact query, while Figure 3 shows the plots for inexact query. Each plot shows the recognition accuracy for a particular matching scheme as a function of the fraction of pattern corruption. The different curves are for the different noise processes. The main feature to note is that in the case of inexact query the shoulder



(a) Ungated Attribute Histogram.

(b) Relational Attribute Histograms.

(c) Feature Sets Matching.

(d) Graph Matching.

Figure 2. Effect of various kinds of noise for exact queries.

in the performance curves occurs at a lower noise level for each of the recognition strategies.

Turning our attention first to the sensitivity curves for exact query, we make the following observations. Firstly, the two histogram based methods (Figure 2(a) and (b)) are most susceptible to the addition of extra line-segments. By contrast, feature-set (Figure 2(c)) comparison and graph-matching (Figure 2(d)) are most susceptible to missing lines.

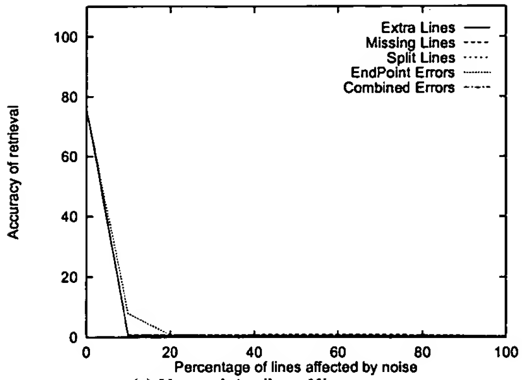
Since the process of inexact query is more typical of large-scale object recognition, we now focus on the associated sensitivity pattern in more detail in Figure 3.

When retrieval is attempted using graph-matching (Figure 3(d)), the following sensitivity pattern emerges. The most destructive noise process is the deletion of lines. Here, the onset of recognition errors occurs when 10% of lines are deleted. Line splitting results in the onset of recognition degradation when the error rate is 20%. The technique is most robust to the addition of extra lines. Here as many as 70% of the lines may be added clutter before any degradation in recognition performance results.

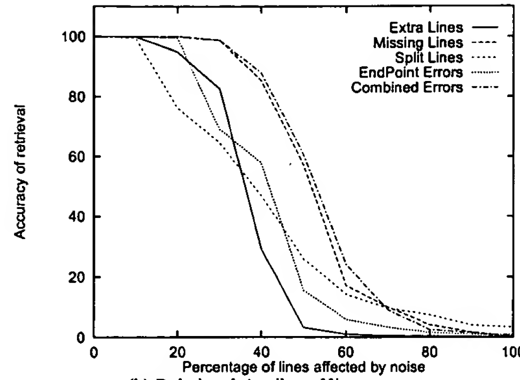
In the case of the feature-sets (Figure 3(c)) it is again line deletion that proves the most serious obstacle to accurate recognition. The onset of errors occurs when 40% of the lines are deleted. The line-patterns are least sensitive to segment end-point errors. In the case of both line-addition and line-splitting there is an onset of errors when the fraction of segment errors is about 20%. However, at larger fractions of segmentation errors the overall effect is significantly less marked than in the case of line-deletions. The technique is considerably less robust to line additions than the graph-matching technique.

In the case of the gated or relational histogram (Figure 3(b)), segment end-point error, missing lines and line-splitting pose the most serious problems. This is attributable to the fact that the line-splitting introduces additional combinatorial background that swamps the query pattern. The line-patterns are least sensitive to segment end-point errors. In the case of both line-addition and line-splitting there is an onset of errors when the fraction of segment errors is about 20%. However, at larger fractions of segmentation errors the overall effect is significantly less marked than in the case of line-deletions. Moreover, in each case the recognition curves are consistently poorer than either graph-matching or feature-sets.

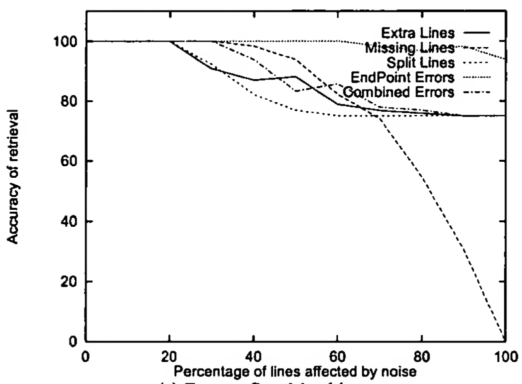
Finally, the recognition performance of the ungated histogram (Figure 3(a)) is poorest of all for inexact pattern matching and is effectively unuseable. Comparing Figures



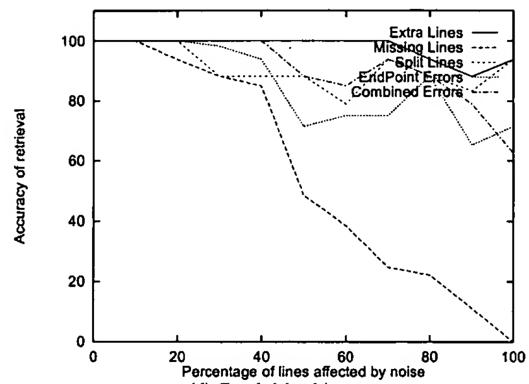
(a) Ungated Attribute Histogram.



(b) Relational Attribute Histograms.



(c) Feature Sets Matching.



(d) Graph Matching.

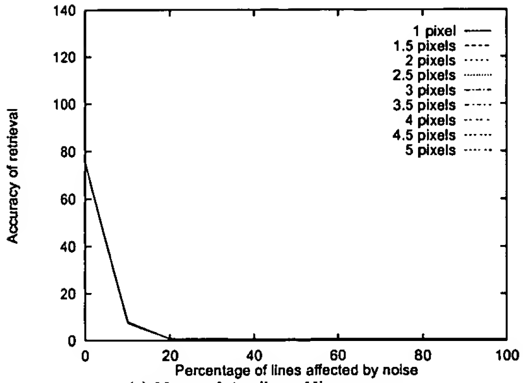
Figure 3. Effect of various kinds of noise for similarity queries.

2a and 3a it is clear that the transition from exact to inexact query has a significant effect on the matching performance.

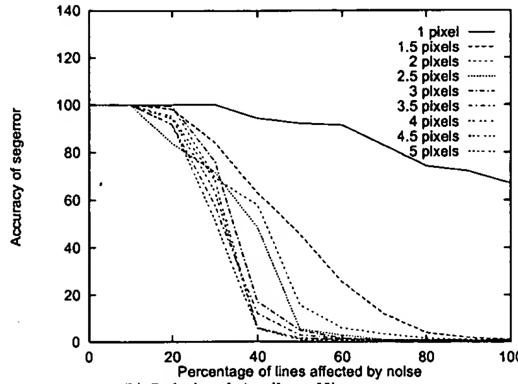
To conclude the sensitivity study, we focus more closely on the role of segment end-point errors. The reason for this is that such errors will effect the accuracy of the relational measurements. Figure 4 shows the effect of line end-point position errors for inexact queries. The different curves in the plots correspond to different values of the standard deviation of the end-point position errors. They show the retrieval accuracy as a function of the fraction of lines affected by end-point errors. As the standard deviation of the position error increases, then so the fraction of corrupt lines for which perfect recall is possible decreases. The main point to note from these plots is that the graph-matching method degrades less rapidly under line end-point errors than the set-based method and the relational histogram.

5 Conclusions

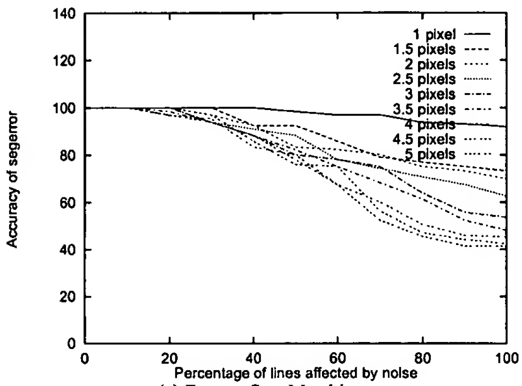
We have compared three different strategies for recalling structural representations of line-patterns from a large database. The main conclusion to be drawn from the study is that the best recognition performance is realised when an iterative graph-matching scheme is used. This is an interesting observation, since graph-structure is frequently perceived as too fragile to be used to noisy information retrieval. There are a number of ways in which the ideas presented in this paper can be extended. Firstly, we intend to explore more a perceptually meaningful representation of the line patterns, using grouping principal derived from Gestalt psychology. Secondly, we are exploring the possibility of using more complex relational entities as alternatives to edges. Examples include the triangular faces of Delaunay graphs [4]. Finally, we plan to investigate ways on combining the recognition strategies. This would offer the advantage that we could take advantage of the complementarity of the relational histogram and graph-matching. The relational histogram is most robust to missing lines while the graph-matching method is most robust to line-deletion. There are two ways in which combination might be achieved. The first is to use the different similarity measures to guide a heuristic search procedure such as the A-star algorithm. The second route is to treat each recognition strategy as a separate classifier and to combine the decisions in an information theoretic framework [11].



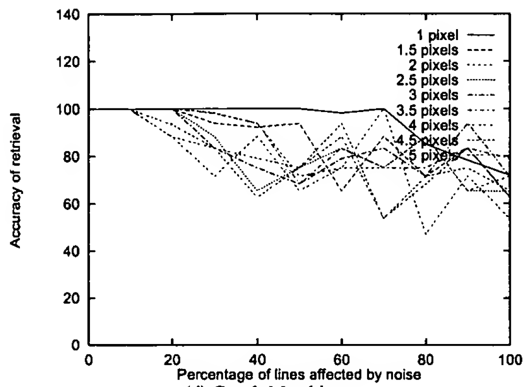
(a) Ungated Attribute Histograms.



(b) Relational Attribute Histograms.



(c) Feature Sets Matching.



(d) Graph Matching.

Figure 4. Effect of introducing segment end-point errors on retrieval performance.

References

[1] Y. Amit and A. Kong. Graphical templates for model registration. *IEEE PAMI*, 18(3):225–236, 1996.

[2] H. Barrow and R. Popplestone. Relational descriptions in picture processing. *Machine Intelligence*, 5:377–396, 1971.

[3] M. Costa and L. Shapiro. Scene analysis using appearance-based models and relational indexing. *ISCV95*, pages 103–108, 1995.

[4] A. M. Finch, R. C. Wilson, and E. R. Hancock. Matching Delaunay graphs. *Patern Recognition*, 30:123–140, 1997.

[5] A. M. Finch, R. C. Wilson, and E. R. Hancock. An energy function and continuous edit process for graph matching. *Neural Computation*, 10:1873–1894, 1998.

[6] J. Henikoff and L. Shapiro. Interesting patterns for model based machine vision. *ICCV90*, pages 535–538, 1990.

[7] B. Huet, A. D. J. Cross, and E. R. Hancock. Graph matching for shape retrieval. *NIPS98*, 1998.

[8] B. Huet and E. R. Hancock. Cartographic indexing into a database of remotely sensed images. *WACV96*, pages 8–14, 1996.

[9] B. Huet and E. R. Hancock. Fuzzy relational distance for large-scale object recognition. *CVPR98*, pages 138–143, 1998.

[10] B. Huet and E. R. Hancock. Relational histograms for shape indexing. *ICCV98*, pages 563–569, 1998.

[11] J. Kittler and S. Hojjatoleslami. A weighted combination of classifiers employing shared and distinct representation. *CVPR98*, pages 924–929, 1998.

[12] M. Leyton. A process grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[13] T. Liu and D. Geiger. Visual deconstruction: recognizing articulated object. *EMMCVPR97*, pages 295–310, 1997.

[14] D. Mumford and J. Shah. Boundary detection by minimizing functionals. *CVPR85*, pages 22–26, 1985.

[15] M. Pelillo *et al.* Matching hierarchical structures using association graphs. *ECCV98*, pages 3–16, 1998.

[16] S. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, vol. 6, pages 730–732, 1977.

[17] W. J. Rucklidge. Locating objects using the Hausdorff distance. *ICCV95*, pages 457–464, 1995.

[18] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graph. *IEEE SMC*, 13:353–362, 1983.

[19] L. Shapiro and R. Haralick. A metric for comparing relational descriptions. *IEEE PAMI*, 7(1):90–94, 1985.

[20] K. Siddiqi *et al.* Shock graphs and shape matching. *ICCV98*, pages 222–229, 1998.

[21] N. Thacker *et al.* Assessing the completeness properties of pairwise geometric histograms. *IVC*, 13(5):423–429, 1995.

[22] R. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE PAMI*, 19(6):634–648, 1997.

[23] S. C. Zhu and A. L. Yuille. Forms: A flexible object recognition and modelling system. *IJCV*, 20(3):187–212, 1996.